

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-265775

(43)Date of publication of application : 15.10.1993

(51)Int.Cl.

G06F 9/46

(21)Application number : 04-063058

(71)Applicant : HITACHI LTD

(22)Date of filing : 19.03.1992

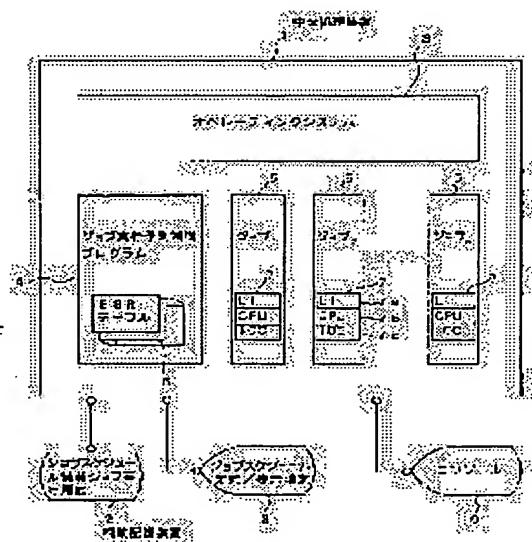
(72)Inventor : HIROZAWA TOSHIO
KURIHARA JUNICHI
ITO TSUTOMU
KUNINISHI MOTOHIDE

(54) JOB EXECUTION FORECAST CONTROL METHOD AND JOB EXECUTION CONDITION DISPLAY METHOD

(57)Abstract:

PURPOSE: To improve the forecast precision of the execution end time of each job regardless of the variation in load on a computer in the information processing system capable of executing plural jobs.

CONSTITUTION: A central processing unit 1 is provided with a job execution forecast control program 4, and this program 4 is provided with plural ESR tables 6 where shedule information of jobs and the execution history are stored. Each job 5 is provided with accounting information 7, and an elapsed time ET 7a, a CPU time 7b, and an I/O frequency 7c are gathered at the time of the end of a job step. At the time of the execution end of each job 5, the processing volume is calculated and is stored in the ESR table 6, and a forecasted end time of the job is displayed on a display terminal 3. If the forecasted end time is later than a desired end time at this time, the execution priority level of this job is raised to approximate the end time to the desired end time.



LEGAL STATUS

[Date of request for examination] 04.11.1998

[Date of sending the examiner's decision of rejection] 27.03.2001

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-265775

(43)公開日 平成5年(1993)10月15日

(51)Int.Cl.⁵

G 0 6 F 9/46

識別記号

3 4 0 A 8120-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数18(全 13 頁)

(21)出願番号 特願平4-63058

(22)出願日 平成4年(1992)3月19日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 廣澤 敏夫

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 栗原 潤一

香川県高松市中央町5番31号 株式会社日

立製作所四国支店内

(72)発明者 伊藤 勉

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 小川 勝男

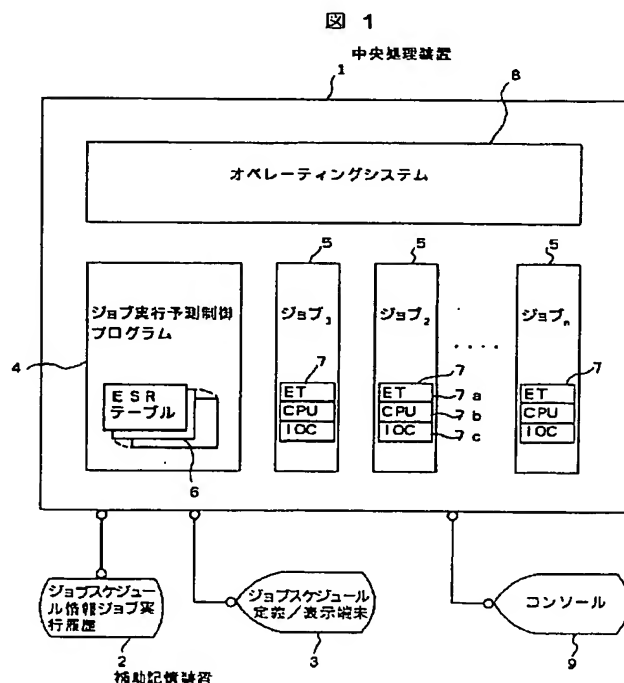
最終頁に続く

(54)【発明の名称】 ジョブ実行予測制御方法およびジョブ実行状況表示方法

(57)【要約】 (修正有)

【目的】複数のジョブが実行できる情報処理システムにおいて、計算機の負荷が変動していても、各ジョブの実行終了時間の予測精度の向上を図る。

【構成】情報処理装置1内に、ジョブ実行予測制御プログラム4を具備し、ジョブ実行予測制御プログラム4内には、ジョブのスケジュール情報と実行履歴を格納する複数個のESRテーブル6がある。各ジョブ5内に課金情報7を有しており、ジョブステップの終了時に経過時間ET7a、CPU時間7b、I/O回数7cを収集する。各ジョブの実行終了時に処理量を計算してESRテーブル6に格納するとともに、表示端末3にジョブの終了予想時刻を表示する。このとき、終了予想時刻が終了希望時刻を越えていたならば、該ジョブの実行優先順位を上げて、終了時刻が終了希望時刻に近づくようにする。



(2)

【特許請求の範囲】

【請求項1】並列に実行される複数のジョブのそれぞれの実行中にそれぞれのジョブの処理量に関する情報を収集し、

該収集された情報に基づき、それぞれのジョブの予想実行終了時刻を計算し、

それぞれのジョブに対して計算された予想実行終了時刻とそれぞれのジョブの希望実行終了時刻とに基づいて、それぞれのジョブのそれ以降の処理スピードを制御するジョブ実行予測制御方法。

【請求項2】いずれかのジョブについて計算された予想実行終了時刻がそのジョブの希望実行終了時刻を越えていたとき、そのジョブのそれ以降の処理のスピードを上げる請求項1記載のジョブ実行予測制御方法。

【請求項3】該処理スピードの制御においては、いずれかのジョブの希望実行終了時間よりも該ジョブの予測実行終了時間が大きい場合に該ジョブの実行優先順位を他のジョブよりも高く設定することにより行うことを特徴とする請求項1記載のジョブ実行予測制御方法。

【請求項4】各実行中のジョブの消費予想CPU時間とそのジョブの消費予想入出力時間を加算して、そのジョブの予想処理量を算出し、

そのジョブの実行に要する希望経過時間により、そのジョブについて算出された予想処理量を割算してそのジョブの処理速度を算出し、

現時点での該情報処理システム単位処理量を計算し、

該ジョブの単位処理量を情報処理システム全体の単位処理量で割算した結果より該ジョブの予想実行終了時刻を計算する請求項1記載のジョブ実行予測制御方法。

【請求項5】該ジョブの消費予想CPU時間と消費予想入出力時間を加算した結果を予想処理量とし、該ジョブの実行終了希望時刻から得るジョブ実行希望経過時間で割算した結果を該ジョブの単位処理量とし、現時点でのジョブの実行が終了したジョブ群の単位処理量を計算し、

該ジョブの単位処理量を現時点でのジョブの実行が終了したジョブ群の単位処理量で割算した結果より該ジョブの予想終了時刻を再計算する請求項1記載のジョブ実行予測制御方法。

【請求項6】前項記載の消費予想入出力時間は、予想入出力回数と単位入出力時間を乗算して得る請求項4または5記載のジョブ実行予測制御方法。

【請求項7】前項記載の単位入出力時間は予想入出力回数の増加に伴い単調減少する関数を用いて計算する請求項4または5記載のジョブ実行予測制御方法。

【請求項8】実行中ジョブについて算出した予測実行終了時間を表示装置に表示する請求項1項記載のジョブ実行予測制御方法。

【請求項9】該処理量収集時に、該実行中ジョブの処理量収集時点までの実行時間と消費CPU時間と入出力発

行回数を収集する請求項1記載のジョブ実行予測制御方法。

【請求項10】該実行中ジョブの処理量を該実行中ジョブのジョブステップ終了時に収集する請求項1記載のジョブ実行予測制御方法。

【請求項11】該ジョブの予想CPU使用時間と予想入出力発行回数を該ジョブの実行前に設定し、

該ジョブの実行時間の予測時に、該ジョブの実行中の処理量と該設定値を使用して該ジョブの実行終了時間を予測する請求項1記載のジョブ実行予測制御方法。

【請求項12】該実行中ジョブの入出力発行回数によって1回当たりの入出力処理時間を決定し、該ジョブの予想入出力回数と乗算することによって、該ジョブ入出力時間を予測する請求項1記載のジョブ実行予測制御方法。

【請求項13】情報処理システム全体の単位処理量は、過去に実行したジョブ群の単位処理量の平均値である請求4または5記載のジョブ実行予測制御方法。

【請求項14】情報処理システム全体の単位処理量は、基準となるジョブの単位処理量である請求項4または5記載のジョブ実行予測制御方法。

【請求項15】それぞれのジョブの実行途中にそれぞれのジョブの実行完了に要する予想実行時間を算出し、その予想実行時間を用いてそのジョブの予想終了時間を算出する請求項1記載のジョブ実行予測制御方法。

【請求項16】並列に実行される複数のジョブのそれぞれの実行中にそれぞれのジョブの処理量に関する情報を収集し、

該収集された情報に基づき、それぞれのジョブの予想実行終了時間を算出し、

それぞれのジョブについて算出した予測実行終了時間をそれぞれのジョブについて指定された希望実行終了時間とをタイムチャート内に表示するジョブ実行状況表示方法。

【請求項17】該収集された情報に基づき、後に実行する予定の複数のジョブの予想実行時間を計算し、それぞれのジョブについて定めた実行開始時刻と算出した予測実行時間とからをそれら後に実行する予定の複数のジョブの予想実行終了時刻を計算し、

それら後に実行する予定の複数のジョブについて指定された希望実行終了時間とそれらのジョブに対して算出された予想実行終了時刻を該タイムチャート内に表示する請求項16記載のジョブ実行状況表示方法。

【請求項18】並列に実行される複数のジョブのそれぞれの実行中にそれぞれのジョブの処理量に関する情報を収集し、

該収集された情報に基づき、後に実行する予定の複数のジョブの予想実行時間を計算し、

それぞれのジョブについて定めた実行開始時刻と算出した予測実行時間とからをそれら後に実行する予定の複数

3

のジョブの予想実行終了時刻を計算し、それら後に実行する予定の複数のジョブについて指定された希望実行終了時間とそれらのジョブに対して算出された予想実行終了時刻をタイムチャート内に表示するジョブ実行状況表示方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は少なくとも1つのジョブが実行可能な計算機システムのジョブの実行制御方法に係り、特に、ジョブの実行前に設定したジョブの実行予定時間通りにジョブを実行する方法に関する。

【0002】

【従来の技術】計算機システムにおけるジョブの実行制御方法は、オペレーティングシステムのマルチプログラミング制御機能の発達にともない複数のジョブを同時に実行する方法が一般的になりつつある。しかし、大量のジョブを一度に実行すると計算機システムの限られた資源に競合が発生し、性能低下を招く恐れがある。この解決方法として、処理特性の似通ったジョブを一つのグループ（ジョブクラス）にまとめ、ジョブクラスを単位にしてジョブの実行を制御する方法が考えられた。この方法によれば、同時に動作するジョブ数は実行するジョブクラスの数で制限することができる。この方法を使用したジョブ実行制御は、「VOS3センタ運営-JSS3偏一、日立製作所HITACマニュアル、8091-3-001」の18ページから24ページに記載されている。しかし、本方法の問題点は、ジョブクラスを設定し、同時に実行するジョブ数を制限したことによって同一クラスのジョブが増加するとジョブクラス内に実行待ちジョブのキューができることになる。この実行待ちジョブキューは、ジョブの投入者にとっては、非常に迷惑なものとなる。すなわち、ジョブ投入者がジョブを投入した時点において実行待ちジョブキューが存在すると、ジョブが何時実行されるのか、何時終了するのか全く検討がつかないことになる。ジョブの中には、所定の時間に実行することが不可欠なジョブも多い。

【0003】この課題を解決するために、ジョブをジョブの実行前に設定したジョブスケジュールに従って実行する方法が考えられた。この方法では、前もってジョブ毎に実行開始時刻を設定しておき、ジョブスケジュール管理プログラムが設定時刻に達したジョブを順番に実行する。この方法は、「統合運用管理システムHOPPS3概説、日立ソフトウエアエンジニアリングHITACマニュアル、」の35ページから36ページに記載されている。本方法によって、ジョブを設定した時間に開始することが可能になった。

【0004】

【発明が解決しようとする課題】しかし、上記ジョブスケジュールを使用したジョブ実行制御方法では、ジョブのスケジューリングにユーザの経験に基づく予想実行時

(3)

4

間を使用していたため、計算機システムの負荷によってジョブの実行時間が異なり、スケジューリングが役立たなくなる。さらに、スケジューリング対象のジョブ群の中で2つ以上のジョブが関連を持つ場合、すなわち、最初に実行したジョブ1が終了した後でないと次のジョブ2が実行できない様な場合は現在時刻がジョブ2に設定した実行開始時刻に達してもジョブ2はジョブ1が終了していなと実行できない。このような関連ジョブが多数ジョブスケジュール中に定義されていると特定のジョブの遅延によってジョブスケジュール全体が狂い、前もって設定したジョブスケジュールが役立たなくなる。

【0005】したがって、本発明の目的は、計算機システムの負荷の変動を考慮に入れて、ジョブの実行時間を予測する方法を提供することにある。

【0006】本発明の他の目的は、計算機負荷の変動がある状態でもジョブをスケジュールに近い形で実行できるようにするジョブ実行制御方法を提供することにある。

【0007】

【課題を解決するための手段】上記目的を達成するために、本発明のジョブ実行予測制御方法では、ジョブのスケジュールリングデータとして、ジョブの開始時刻と終了予定時刻に加え、ジョブの予想CPU使用時間と予想入出力回数を定義する。ジョブの実行が開始されてからは、スケジュール対象ジョブのジョブステップ終了時点において、ジョブの処理履歴として、そのジョブの現時点までの実行時間とCPU使用時間と入出力発行回数を収集した後、この処理履歴を使用してそのジョブの処理スピードである実行サービス率ESRを計算し、そのESRを使用してジョブ毎の未実行部分、すなわち、スケジュールデータで定義した予想CPU使用時間と予想入出力回数から処理履歴として収集したのCPU使用時間と入出力発行回数を除いた部分の実行時間を予測する。この結果、スケジュール対象ジョブの終了予測時刻がスケジュールで設定した終了予定時刻を越える場合は、そのジョブの実行優先順位を上げるか、あるいは他のジョブの実行を一時的に中断させることによって、スケジュール通りにジョブが実行されるように制御する。なお、ESRの定義に関しては、下記の刊行物を参照されたい。

【0008】「計算機システムにおける性能管理の一方式とそれをういた実験」、情報処理学会論文誌、第23巻第6号591頁-598頁

【0009】

【作用】本発明のジョブ実行予測制御方式は、計算機システム内で一つのジョブとして動作するため、ジョブのスケジュール制御を行わない場合は動作しないので、従来のジョブ実行制御方式に対して誤操作させることはない。

【0010】

50

5

【実施例】以下、本発明の一実施例を図1から図11により説明する。図1は、本発明のジョブ実行予測制御方法を適用した計算機システムの構成を表わしている。本発明のジョブ実行予測制御方法は、従来の計算機システムの構成要素である主記憶装置を含む中央処理装置1と中央処理装置1で動作する少なくとも1つ以上のジョブ5を同時に実行させうるオペレーティングシステム(OS)8、オペレーティングシステム8を操作するコンソール9、ジョブのスケジュール情報とジョブの実行履歴を格納する補助記憶装置2、ジョブのスケジュール定義と表示を行う端末3、本発明のジョブ実行予測制御方法を具備したジョブ実行予測制御プログラム4で構成している。

【0011】本実施例では、本発明のジョブ実行予測制御方法を実現するジョブ実行予測制御プログラム4はオペレーティングシステム8の制御下で1つのジョブとして動作するが、本プログラム4はオペレーティングシステム8に組み込まれても良い。ジョブ実行予測制御プログラム4には、ジョブのスケジュール情報と実行履歴を格納する複数のテーブル(ESRテーブル)6を具備している。各々のジョブ5は、ジョブの課金情報7を具備する。課金情報7は、ジョブの実行を開始してからの経過時間(ET)7a、使用したCPU時間(CPU)7b、入出力発行回数(IOC)7cを格納している。

【0012】図2は、本発明のジョブ実行予測制御プログラム4の詳細な構成を示している。ジョブ実行予測制御プログラム4は、ジョブスケジューリング部10、ジョブ実行結果表示部16、ESR算出部12、ジョブ起動処理部13、ジョブ実行制御部14、ジョブ実行履歴情報収集部15で構成する。ジョブ実行予測制御プログラム4には、ジョブのスケジュール情報と実行履歴を格納する複数のESRテーブル6を具備している。

【0013】図3は、ESRテーブル6の構成を示している。ESRテーブル6は、ジョブの実行状態を表す状態フラグ21、スケジュール対象のジョブ名称を格納する領域22、ジョブを規定の時間内に終了させるための実行スピードを表わす指標である目標ESR(Eo)を格納する領域23、実際にジョブを実行した際の実行スピードである実現ESR(Er)を格納する領域24、ジョブの開始予定時刻(Ts)を格納する領域25、ジョブの終了予定時刻(Te)を格納する領域26、ジョブのスケジュール定義時に初期設定したジョブのCPU時間(Ci)を格納する領域28、ジョブのスケジュール定義時に設定したジョブのI/O発行回数(Ii)を格納する領域29、ジョブの実行を開始してからの経過時間(Tc)を格納する領域30、ジョブの実行を開始してからの消費CPU時間(Ce)を格納する領域31、ジョブの実行を開始してからの発行済みI/O回数(Ie)を格納する領域32、該ジョブを実行させるために実行が終了している必要のあるスケジュール対象ジ

(4)

6

ョブ中の関連ジョブ名称を格納する領域33で構成する。

【0014】図4は、ジョブのスケジュール定義表である。ジョブのスケジュール情報として、そのジョブについて、その名称35a、予想CPU時間35b、予想I/O回数35c、ジョブの開始予定時刻35dと目標終了時刻35e、そのジョブの実行開始前に終了していなければならないジョブ(優先ジョブ)のリスト35fを定義する。本表の定義には、図1のジョブスケジュール定義/表示端末3を使用する。開示の例では、JOB1からJOB6までの6本のジョブのスケジュールを定義している。JOB1は18:00に実行を開始し、19:00までに終了させる。同様に、JOB2は18:00に実行を開始し、19:30までに終了させる。JOB3は19:00に実行を開始し、20:00までに終了させる。ただし、JOB3はJOB1の実行が終了した後でないと実行できない。同様に、JOB4はJOB2、JOB5はJOB3とJOB4、JOB6はJOB5が終了した後でないと実行できない。

【0015】図5は、図4のジョブスケジュール情報をもとに後述の計算により求めた結果を示すチャートである。横軸は時刻を表している。例えば、18は18時00分を表す。図中の*記号は図4のジョブスケジュール表で定義したジョブの目標終了時刻を表わし、*記号はジョブ実行予測制御プログラム4が現在の計算機の負荷状態から予想したそのジョブの未実行部分の予想実行時間を表わしている。1つの*記号が10分に相当する。このタイムチャートはジョブ実行予測制御プログラム4によって図1のジョブスケジュール定義/表示端末3に表示される。

【0016】図6は、図5と同様にジョブのタイムチャートを表わしている。図5との違いは、図6ではジョブの実行後の結果を同時に表示することにある。ここでは、JOB1とJOB2が終了した時点でJOB3からJOB6の各々の予想実行時間を再計算して表示している。図中の@記号がジョブの実行結果である。JOB1の実行に60分かかり、JOB2の実行に90分かかっている。ここで、JOB4はJOB2が終了した後でないと実行を開始出来ないため(図4参照)、JOB4の予想実行時間を再計算した結果、JOB4はJOB4の目標終了時刻20:00を超えることになる。本発明の第2の特徴は、ジョブの実行時間を予測した結果がジョブの目標終了時刻を超える場合に、そのジョブの処理速度を向上させて目標終了時刻に近づける制御方法を実現している。この制御方法は後に説明する。

【0017】以下に本実施例の処理の詳細を示す。図7から図11は、図2のジョブ実行予測制御プログラム4の主要部の処理フローである。図7はジョブスケジューリング部の処理フローを表わし、図8はジョブ実行結果表示部11の処理フロー、図9はESR算出部12の処

(5)

7

理フローを示す。また、図10はジョブ起動処理部13の処理フロー、図11はジョブ実行制御部14で該当ジョブの優先順位を上げる処理の処理フローを示す。

【0018】本実施例では、ジョブ実行予測制御プログラム4は図1のコンソール9からOSコマンドであるSTARTコマンドを投入することによって1つのジョブとして起動される。ジョブ実行予測制御プログラム4が起動されるとオペレーティングシステム8から最初にジョブスケジューリング部10に制御が渡る。

【0019】図7を参照するに、ジョブスケジューリング部10に制御が渡ると処理ステップ41aにて、図4に示す各ジョブのスケジューリングデータ35を図1、または図3に示した、そのジョブのESRテーブル6に格納する。すなわち、ジョブスケジューリングデータ35の各ジョブのジョブ名称35aはジョブ名称領域22へ、CPU時間35bは初期設定CPU時間領域28へ、I/O回数35cは初期設定I/O発行回数領域29へ、ジョブ開始時刻35dはジョブ開始時刻領域25へ、ジョブ終了時刻35eはジョブ終了時刻領域26に格納する。

【0020】次の処理ステップ41bは、ESRテーブル6の実現ESR領域24に初期値としてシステムESR値を格納する。

【0021】各ジョブのESRは、そのジョブの処理速度を示す指標であり、数1で定義される。このESRについては、前述の情報処理学会論文誌を参照されたい。

【0022】

CPU時間+入出力時間

ESR = ----- (数1)

経過時間

入出力時間は、1回の入出力に要する平均入出力時間とジョブスケジュール表35の入出力回数35cの積で算出する。従来のESRの計算方法では、1回当たりの平均入出力時間に固定値(30ミリ秒)を使用していた。本発明の第1の特徴は、数1に示す入出力時間の計算式に入出力の発行回数(I/O発行回数)によって1回当たりの平均入出力時間を変化させることにある。本実施例では、平均入出力時間の算出関数として、実測データから求めた数2を採用している。

【0023】

$f(X) = (A-B) \cdot \exp(-\alpha \cdot \log(X/X_0))$ (数2)

数2において、XはジョブのI/O発行回数、 X_0 はジョブの最小I/O発行回数、AはI/O発行回数が X_0 回の時のI/O時間、BはI/O発行回数が無限大の時のI/O時間、 α は減少係数、を表わす。本実施例では X_0 を100回、Aを100ミリ秒、Bを5ミリ秒、 α を2.3に設定した。なお、本設定値はジョブの特性に依存した値を設定可能である。

【0024】本実施例では、システムESRとして、過去に実行済のジョブ実行履歴から計算した各ジョブのE

8

SRの平均値を使用する。その他の方法として、基準となるジョブを常時動作させ、その実現ESR値を使用する方法でも良い。

【0025】処理ステップ41cでは、目標ESRを算出するために、ESR算出処理部12をコールする。目標ESRは、そのジョブを開始予定の時刻から終了予定の時刻までの予定経過時間内で終了させるに必要なESRである。

【0026】図9はESR算出部12の処理フローであり、図7の処理ステップ41cと図8の処理ステップ51からコールされる。処理ステップ60では、数2を使用し、平均I/O時間(IOT)を計算する。処理ステップ61では、数1に基づいてESRを計算する。ESR算出部12は、図7の処理ステップ41cでコールされた場合、目標ESR(E_o)を計算するために、CPU時間には、そのジョブのESRテーブル内の初期設定CPU時間(C_i)28を使用し、I/O発行回数には初期設定I/O発行回数(I_i)29とし、経過時間にはジョブ終了予定時刻(T_e)30とジョブ開始予定時刻(T_s)25との差を使用する。また、実現ESR(E_r)24を計算する場合は、CPU時間に消費CPU時間(C_e)31を使用し、I/O時間に発行済みI/O回数(I_e)32を使用する。次に、計算されたESRを目標ESR領域23に格納する。図4の例では、JOB1を18:00に実行させ、19:00に終了させるための目標ESRは0.10となる。同様にJOB2を19:30に終了させるための目標ESRは0.11となる。

【0027】処理ステップ41cでは、次にそのジョブの予想実行時間(T_n)を計算し、その結果を端末3に表示する。その表示例が図5である。図5の中の*記号で示すジョブの予想実行時間は、予測時点でのジョブの実現可能なESR(これをシステムESRと呼ぶ)を0.2と仮定し計算した。したがって、JOB1の予想実行時間(T_n)は数3から30分となる。

【0028】

$T_n = (C_i + f(I_i) \times I_i) / \text{システムESR}$ (数3)

JOB1と同様に、JOB2からJOB6に関しても数12を使用して予想実行時間(T_n)を算出し、結果を表示する。すなわち、これは図7の処理ステップ41aから処理ステップ41cを図4に登録された個数分(n回)を繰返せば良い。なお、図7では、この初期状態での各ジョブの予想実行時間を計算し、表示するための繰返し処理の処理フローは省略してある。

【0029】処理ステップ41eにてカウンタiを1に初期化する。すなわち、処理ステップ41aから処理ステップ41eによって、ESRテーブル6の初期設定が完了する。

【0030】以後、全てのジョブの実行が終了するまで処理ステップ41fから処理ステップ41nを繰返し実

(6)

9

行する。ここで、図7の記号nはスケジュール対象ジョブ数、iはn個のジョブ中のi番目のジョブを示している。ジョブスケジューリング部10の処理フロー図7の処理ステップ41fには、各ジョブのジョブステップが終了したときに、必ずオペレーティングシステム8から制御が渡る。すなわち、図2で示したジョブ実行履歴情報収集部15より制御が渡る。このジョブ実行履歴情報収集部15より制御が渡る方法については、後に説明する。

【0031】ステップ41fでは、図5に例示した示すジョブ実行時間を予測したタイムチャートを表示するためにジョブ実行予測結果表示部16（図2）をコールする。

【0032】処理ステップ41fからコールされる実行予測結果表示部16では、図8に示すように、処理ステップ50にてi番目のESRテーブル6をポイントする。処理ステップ51では、i番目のジョブの実現ESRを計算するためにESR算出部12をコールする。ESR算出処理部12は、この処理ステップ51からコールされた場合には、実現ESRを計算するために、CPU時間に消費CPU時間（Ce）31を使用し、I/O時間に発行済みI/O回数（Ie）32を使用して図9に従いESRを算出する。処理ステップ52では、処理ステップ51で得られた実現ESR（Er）24と初期設定CPU時間（Ci）28、初期設定I/O発行回数（Ii）29、消費CPU時間（Ce）31、発行済みI/O回数（Ie）32から未実行部分の予想実行時間（Tx）を数4にて計算する。

【0033】 $T_x = ((C_i - C_e) + (I_i - I_e) * f(I_i)) / ESR$ （数4） 処理ステップ53、処理ステップ54では、ジョブの実行を開始してから現在までの経過時間（Tc）を@記号で端末3に表示し、未実行部分の予想実行時間（Tx）を*記号で表示する。図6がその表示例である。なお、Ce、Ieとも0であるならば、図5に示すように、@記号は表示されない。次の処理ステップ55にて、現在時刻がi番目のジョブの実行終了予定開始時刻を過ぎていると判断された場合に、そのジョブの実行スピードを上げるためにジョブ実行制御部14をコールし、ジョブの実行優先順位を100から101に変更する。この処理の詳細は、第7図の処理ステップ41jで説明する。

【0034】処理ステップ41gと処理ステップ41hでは、現在時刻とi番目のジョブの実行開始時刻を比較し現在時刻がそのジョブの実行開始時刻を越えており、かつそのジョブに対する優先ジョブの実行が全て終了しているならば、そのi番目のジョブを起動するために処理ステップ41iにてジョブ起動処理部13（図2）をコールする。図10は、ジョブ起動処理部13の処理フローを表している。図10において、ジョブ起動処理部13がコールされると、最初に処理ステップ70にて、

10

i番目のジョブiが既に起動済みか否かをチェックする。ジョブの状態（実行前、実行中、実行終了）はESRテーブル6の状態フラグ21（図3）で管理する。

【0035】一方、ジョブiが動作中でないならば、処理ステップ71にて疑似コンソールをオープンする。疑似コンソールは、ジョブ実行制御プログラム4からオペレーティングシステム8に対して、各種の制御コマンドを発行するためのインタフェースであり、詳細に関しては下記の刊行物に開示されている。

【0036】「VOS3 システムプログラマの手引 - マクロ編 - 日立製作所 HITACマニュアル」（資料番号6180-3105, 285-299ページ）

処理ステップ72では、図7のジョブスケジューリング部10の処理ステップ41iで指定されたジョブiのジョブ名称をパラメータにしたジョブ起動コマンド（≠Sコマンド）を発行することにより、オペレーティングシステム8によってジョブiが起動される。ジョブiが起動された後は、処理ステップ72で疑似コンソールをクローズする。

【0037】再び図7参照するに、処理41gにおいて、現在時刻がi番目のジョブの実行開始時刻を越えていると判断されたが、処理ステップ41hにおいて、そのジョブの優先ジョブの実行が終了していないと判断された場合は、その優先ジョブの実行スピードを上げるために処理41jにてジョブ実行制御部14をコールし、ジョブの実行優先順位を100から101に変更する。

【0038】図11は、ジョブ実行制御部14の当該ジョブの優先順位を上げる処理の処理フローである。処理ステップ80では、ジョブステップの実行優先順位を変更することを指示するCHAPマクロを使用し、指定されたジョブを構成しているジョブステップ群の中で現在実行中のジョブステップの実行優先順位を変更する。CHAPマクロの詳細は、「VOS3 システムプログラマの手引 - マクロ編 - 日立製作所 HITACマニュアル」（資料番号6180-3105, 78-79ページ）に開示されている。

【0039】これにより、図6のJOB4に示すように、ジョブの実行時間を予測した結果がジョブの目標終了時刻を超える場合に、そのジョブの処理速度であるESRを向上させることになり、結果としてそのジョブの実行終了時刻が早まる。

【0040】以上の処理が全てのジョブについて繰返し行われる（処理ステップ41fから処理ステップ41Lまで）。

【0041】以上の処理ののち、処理ステップ41mにてカウンタiを1に初期化した後、処理ステップ41nにて、図4で示したスケジュール対象のジョブが全て終了したかを調べる。終了していないならば、処理ステップ41fへ戻り、全て終了していたならば、本発明のジ

(7)

11

ジョブ実行予測制御方法を実現している処理プログラム4が終了する。

【0042】起動可能なジョブがないとき、あるいは予定開始時刻に至ったジョブがないときには、すでに実行中のジョブに対して、その実行途中の経過をみて予想終了時刻の再計算を行う。

【0043】その説明の前にジョブ実行履歴収集部15の説明をする。ジョブ実行履歴収集部15は、スケジュール対象のジョブのジョブステップ終了時にオペレーティングシステム8から制御が渡る。一般的に1つのジョブは、少なくとも1つ以上のジョブステップで構成されている。

【0044】ジョブ実行履歴情報収集部15は、スケジュール対象ジョブの1つのジョブステップが終了する度に起動され、図1で示したジョブの課金情報7からジョブの経過時間と消費CPU時間と発行したI/O回数を取得してジョブ対応に用意しているESRテーブル6に格納する。なお、消費CPU時間やI/O回数の収集は、特許公報特公平1-39133号公報に開示されている方法を用いても良い。さて、ジョブ実行制御部14では、前述の起動処理のあと、実行中のジョブに対して予想終了時刻を再計算する。すなわち、図7において、実行中のジョブがあるときには（処理ステップ41n）、再度処理ステップ41fを起動し、ジョブ実行予想表示部16をコールする。この表示部16は、すでに説明したように、図8の処理を行う。図8の処理は先に述べた通りであり、その結果図6で示した情報が端末3に常時表示される。

【0045】以上が本発明のジョブ実行予測制御方式の一実施例である。本実施例では、図4に示すジョブのスケジュールデータを端末3から入力しているが、前もって補助記憶装置2に格納しあるスケジュールデータを入力しても良い。

【0046】

【発明の効果】本発明によれば、ジョブのCPU時間とI/O発行回数と経過時間を定期的にモニタリングし、ジョブの実行スピードを表わす指標であるESRを計算することによりジョブの未実行部分の実行時間を予測でき、計算機システムの負荷に依存して変動するジョブの終了時刻をジョブ投入者に教えることが可能になり、従来の経験的なジョブ終了予定時刻を明確化できる効果が

12

有る。

【0047】さらに、ジョブの実行時間を予測することにより、ジョブの目標終了時刻よりも予想終了時刻が遅れた場合は、何らかのジョブの処理スピードの向上手段を実施でき、前もって設定したジョブのスケジュールの中で、特定のジョブの実行の遅延による他のジョブへの影響を従来に比べ局所化できる効果がある。

【図面の簡単な説明】

【図1】本発明のジョブ実行予測制御方式を具備した計算機システムの構成図である。

【図2】ジョブ実行予測制御プログラムの構造を示した図である。

【図3】ジョブの実行予測データを格納するESRテーブルの構造を示した図である。

【図4】ジョブスケジュールデータの定義例の例を示した図である。

【図5】ジョブ実行予測結果のタイムチャートを示した図である。

【図6】ジョブ実行予測結果のタイムチャートを各ジョブが終了した時点ごとに再計算して示した図である。

【図7】ジョブスケジューリング部の処理フローを示した図である。

【図8】ジョブ実行結果予測制御部の処理フローを示した図である。

【図9】ESR算出部の処理フローを示した図である。

【図10】ジョブ起動処理部の処理フローを示した図である。

【図11】ジョブ実行制御部におけるジョブの実行優先順位を上げる処理の処理フローを示した図である。

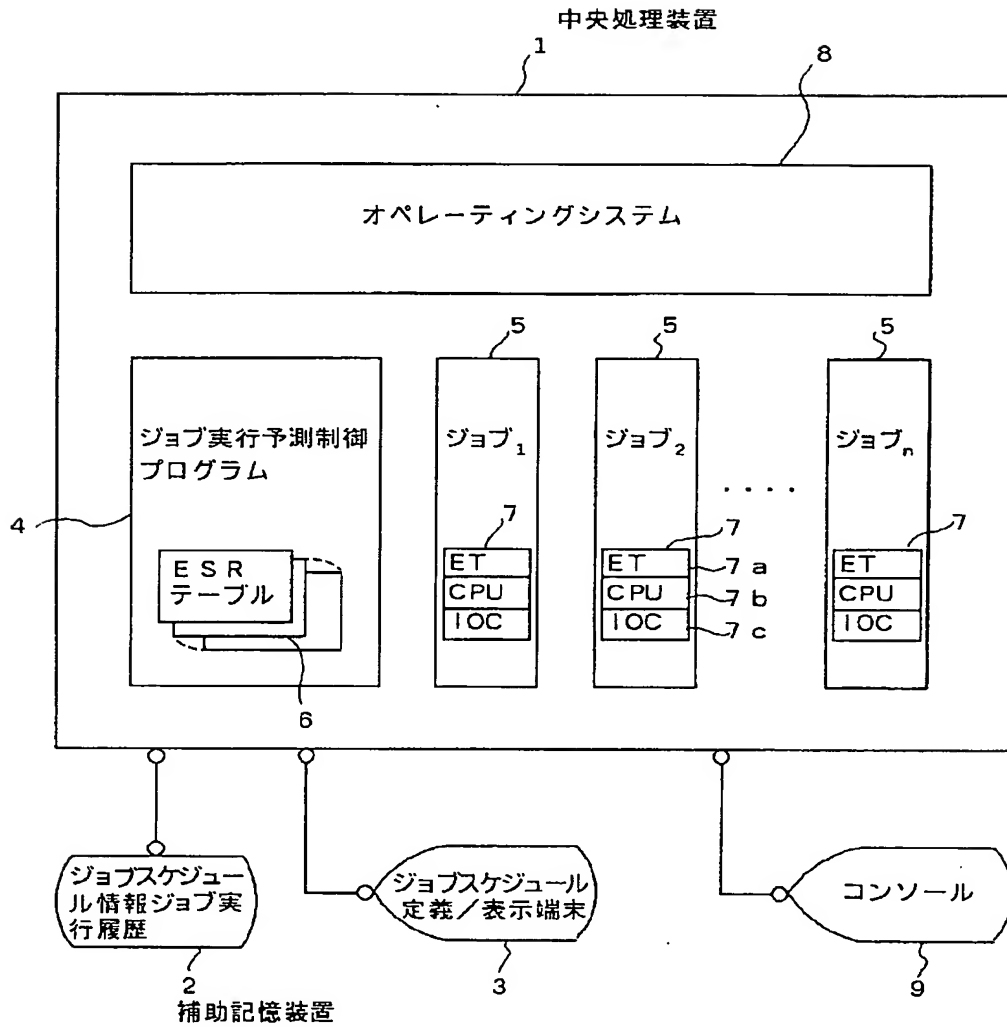
【符号の説明】

1・・・中央処理装置、2・・・補助記憶装置、3・・・ジョブスケジュール表示／制御端末、4・・・ジョブ実行予測制御プログラム、5・・・ジョブ、6・・・ESRテーブル、7・・・課金情報、8・・・オペレーティングシステム、9・・・コンソール、10・・・ジョブスケジューリング部、16・・・ジョブ実行予測結果表示部、12・・・ESR算出部、13・・・ジョブ起動処理部、14・・・ジョブ多重度制御部、15・・・ジョブ実行履歴情報収集部、20・・・ジョブスケジュール表。

(8)

【図1】

図 1



【図5】

図 5

	18	19	20	21	22	目標ESR
JOB 1	***	≡				0.10
JOB 2	*****	≡				0.11
JOB 3		***	≡			0.08
JOB 4		**	≡			0.11
JOB 5		***		≡		0.07
JOB 6			**		≡	0.09

*予測実行時間 ≡ : 目標終了時刻

【図6】

図 6

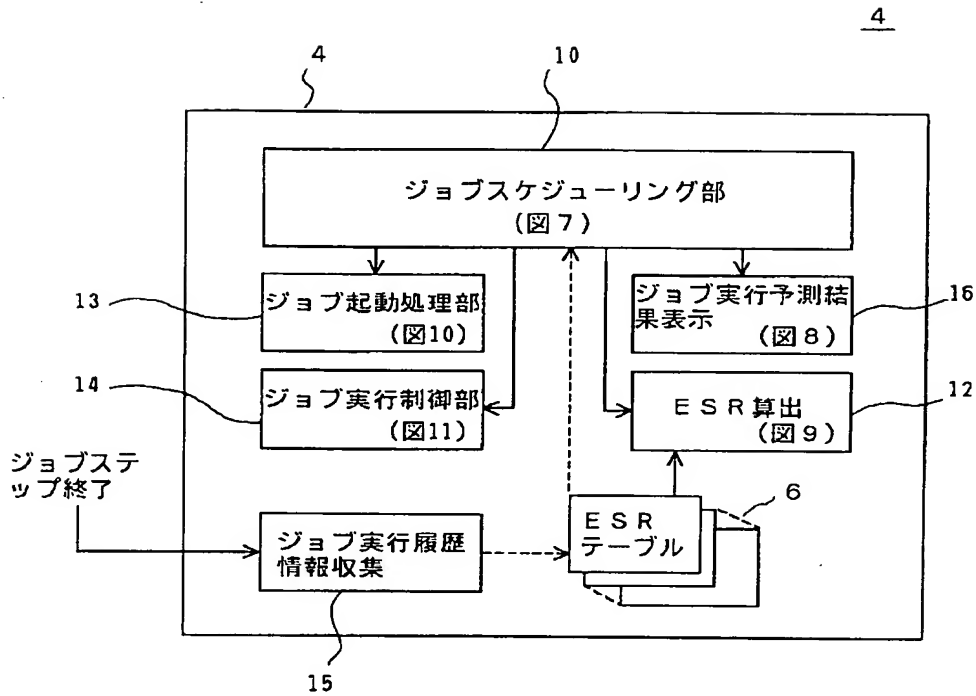
	18	19	20	21	22	目標ESR
JOB 1	⊗⊗⊗⊗⊗⊗					0.10
JOB 2	⊗⊗⊗⊗⊗⊗⊗⊗					0.10
JOB 3		*****				0.08
JOB 4		***				0.11
JOB 5			*****			0.07
JOB 6			**	***	≡	0.09

*予測実行時間 ≡ : 目標終了時刻 ⊗ : 実行結果

(9)

【図2】

図 2



【図4】

図 4

3.5 ジョブスケジュール表

35 a ジョブ名	35 b CPU時間	35 c I/O回数	35 d 開始予定時刻	35 e 終了予定時刻	35 f 優先ジョブリスト	
JOB 1	200	10000	18:00	19:00		
JOB 2	300	15000	18:00	19:30		
JOB 3	100	20000	19:00	20:00	JOB 1	
JOB 4	50	5000	19:30	20:00	JOB 2	
JOB 5	100	10000	20:00	21:00	JOB 3	JOB 4
JOB 6	10	5000	21:00	21:30	JOB 5	

(10)

【図3】

図 3

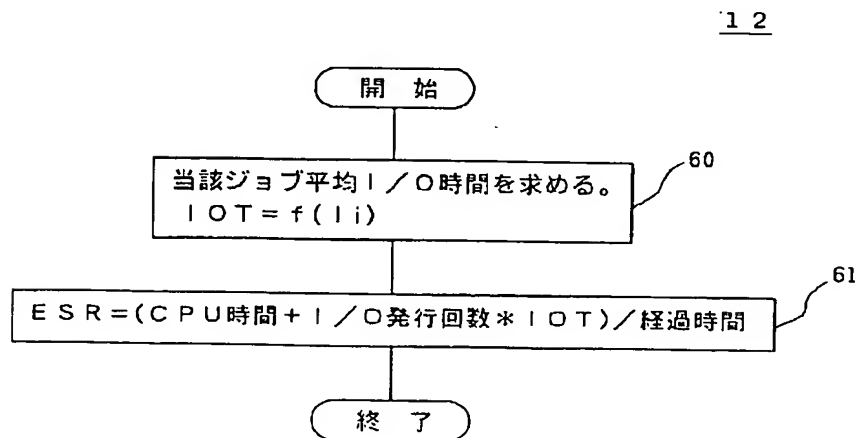
6 ESRテーブル

21

+ 0	フ ラ グ		
+ 4	ジ ョ ブ 名 称		
23			
+ 12	目 標 E S R (E o)	実 現 E S R (E r)	
+ 16	ジ ョ ブ 開 始 予 定 時 刻 (T s)		
+ 20	ジ ョ ブ 終 了 予 定 時 刻 (T e)		
+ 24	初 期 設 定 C P U 時 間 (C l)		
+ 28	初 期 設 定 I / O 発 行 回 数 (l i)		
+ 32	経 過 時 間 (T c)		
+ 36	消 費 C P U 時 間 (C e)		
+ 40	発 行 済 み I / O 回 数 (l e)		
+ 44	優 先 ジ ョ ブ 名 称 1		
+ 48	優 先 ジ ョ ブ 名 称 2		

【図9】

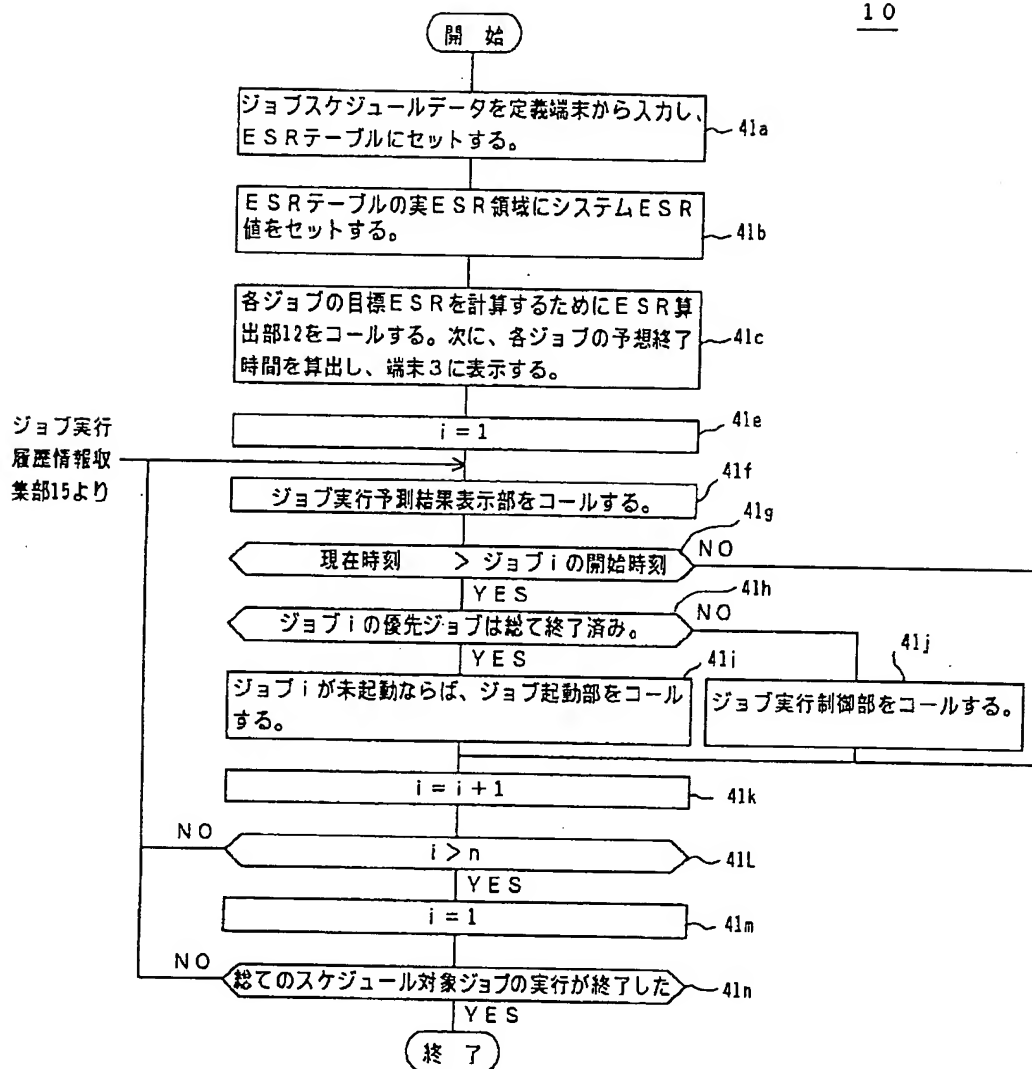
図 9



(11)

【図7】

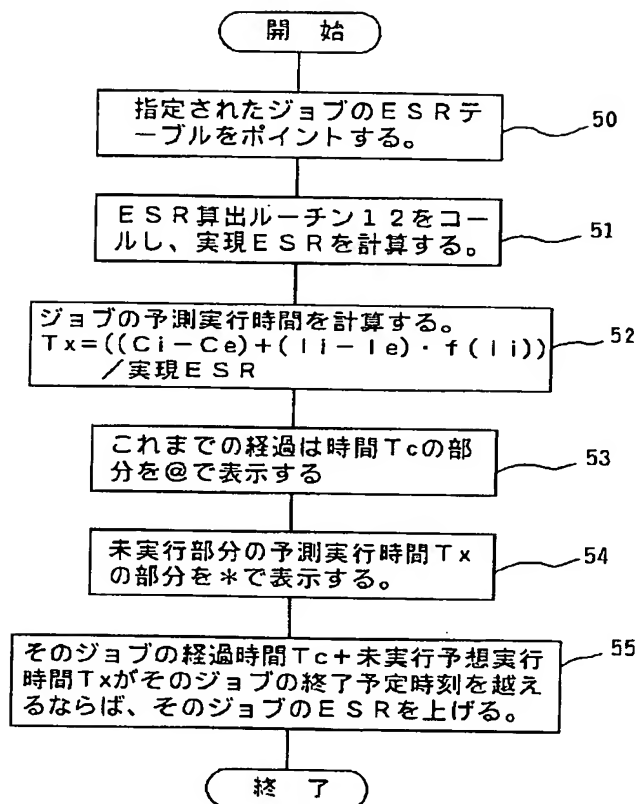
図 7

10

(12)

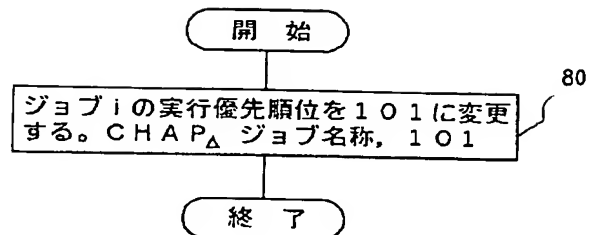
【図8】

図 8



【図11】

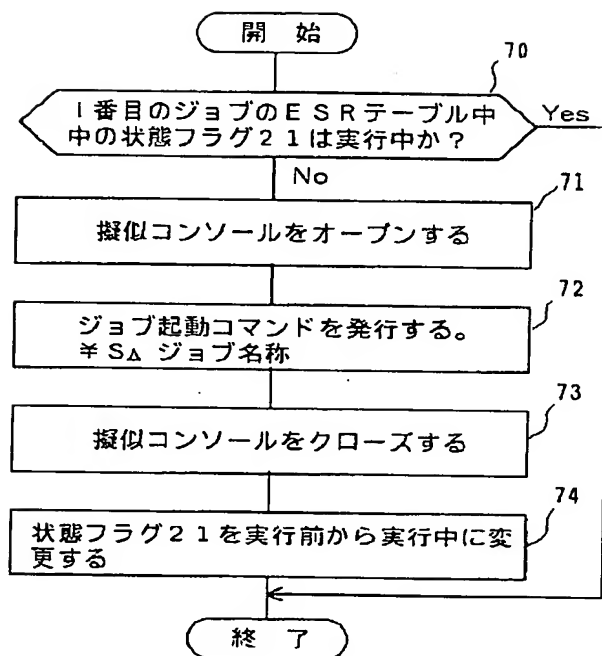
図 11



(13)

【図10】

図 10

13

フロントページの続き

(72) 発明者 国西 元英

東京都国分寺市東恋ヶ窪1丁目280番地
株式会社日立製作所中央研究所内